

ISIS-II PL/M-86 V2.0 COMPILATION OF MODULE LMCMD

OBJECT MODULE PLACED IN LMCMD.OBJ

COMPILER INVOKED BY: :F0: LMCMD.PLM DATE( MARCH 16, 1982 ) XREF OPTIMIZE(3) DEBUG

```

1      LMCMD:
      DO;
      /* CP/M 8086 CMD file generator from output of Intel Locator

```

```

      COPYRIGHT (C) 1982
      DIGITAL RESEARCH
      BOX 579 PACIFIC GROVE
      CALIFORNIA 93950

```

```

      */

```

```

2 1    DECLARE
      paragraph literally '16',
      ex literally '12',          /* extent */
      nr literally '32',          /* current record */
      maxb address external,
      fcb(33) byte external,      /* DEFAULT FILE CONTROL BLOCK */
      buffa(128) byte external; /* DEFAULT BUFFER ADDRESS */

```

```

3 1    DECLARE COPYRIGHT(*) BYTE DATA
      (' COPYRIGHT (C) 1982, DIGITAL RESEARCH ');

```

```

4 1    MON1: PROCEDURE(F,A) EXTERNAL;
5 2    DECLARE F BYTE, A ADDRESS;
6 2    END MON1;

```

```

7 1    MON2: PROCEDURE(F,A) BYTE EXTERNAL;
8 2    DECLARE F BYTE, A ADDRESS;
9 2    END MON2;

```

```

10 1   DECLARE SP ADDRESS;

```

```

11 1   BOOT: PROCEDURE;
12 2   call mon1 (0,0);
13 2   END BOOT;

```

```

14 1   declare segmts(11) structure (name(5) byte)
      initial ('CODE ', 'DATA ', 'EXTRA', 'STACK',
        'X1 ', 'X2 ', 'X3 ', 'X4 ', '8080 ', 'NZERO',
        'NHEAD');

```

```

15 1   declare group (*) byte initial ('CGROUP', 'DGROUP', 'EGROUP', 'SGROUP');

```

```

16 1   declare group$index(4) byte initial (0,0,0,0);

```

```

17 1   declare lseg$length(16) address initial (0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0); /* segment lengths from x
      record */

```

```

18 1   declare header (15) structure
      (typseg byte, file$length address, absolute$add address,
      minimum$mem address,

```

CP/M-86 v.1.1 (Vol. II)  
 COPYRIGHT © 1981, 1982  
 DIGITAL RESEARCH  
 P. O. BOX 579  
 PACIFIC GROVE, CA 93950  
 SER. # C81-162

```
maximum$mem address) initial {0,0,0,0,0,0,0,0,0,0,0,0,0,0,
```

```

17 1      pla: PROCEDURE public;
20 2          DECLARE FCB (33) BYTE AT (.FCBA),
           DFCBA LITERALLY 'FCBA';
21 2          DECLARE BUFFER (128) BYTE AT (.BUFFA),
           DBUFF LITERALLY 'BUFFA';
22 2          DECLARE SFCB(33) BYTE, /* SOURCE FILE CONTROL BLOCK */
           BSIZE LITERALLY '1024',
           EOFILE LITERALLY '1AH',
           SBUFF(BSIZE) BYTE, /* SOURCE FILE BUFFER */
           RFLAG BYTE,        /* READER FLAG */
           SBP ADDRESS;        /* SOURCE FILE BUFFER POINTER */
23 2          declare tbp address; /* pointer to command tail */
24 2          declare count$command$tail byte at (.buffa);
25 2          declare (t8080,nozero) byte;

```

```
26 2 DECLARE
      TRUE LITERALLY '1',
      FALSE LITERALLY '0',
      FOREVER LITERALLY 'WHILE TRUE',
      CR LITERALLY '13',
      LF LITERALLY '10',
      WHAT LITERALLY '63';
```

```

27  2      patch: procedure;
28  3      declare i byte;

```

```

29 3          i = i + 0;
30 3          i = i + 0;
31 3          i = i + 0;
32 3          i = i + 0;
33 3          i = i + 0;
34 3          i = i + 0;
35 3          i = i + 0;
36 3          i = i + 0;
37 3          i = i + 0;
38 3          i = i + 0;
39 3          i = i + 0;
40 3          i = i + 0;
41 3          i = i + 0;
42 3          i = i + 0;
43 3          i = i + 0;
44 3          i = i + 0;
45 3          i = i + 0;
46 3          i = i + 0;
47 3          end patch;

```

```
48 2 PRINTCHAR: PROCEDURE(CHAR);
49 3 DECLARE CHAR BYTE;
50 3 CALL NON1(2,CHAR);
```

COPYRIGHT © 1981, 1982  
DIGITAL RESEARCH  
P. O. BOX 579  
PACIFIC GROVE, CA 93950

```
51 3      END PRINTCHAR;

52 2      CRLF: PROCEDURE;
53 3          CALL PRINTCHAR(CR);
54 3          CALL PRINTCHAR(LF);
55 3      END CRLF;

56 2      PRINTNIB: PROCEDURE(N);
57 3          DECLARE N BYTE;
58 3          IF N > 9 THEN CALL PRINTCHAR(N+'A'-10); ELSE
60 3              CALL PRINTCHAR(N+'0');
61 3      END PRINTNIB;

62 2      PRINTHEX: PROCEDURE(B);
63 3          DECLARE B BYTE;
64 3          CALL PRINTNIB(SHR(B,4)); CALL PRINTNIB(B AND 0FH);
66 3      END PRINTHEX;

67 2      PRINTADDR: PROCEDURE(A);
68 3          DECLARE A ADDRESS;
69 3          CALL PRINTHEX(HIGH(A)); CALL PRINTHEX(LOW(A));
71 3      END PRINTADDR;

72 2      PRINTM: PROCEDURE(A);
73 3          DECLARE A ADDRESS;
74 3          CALL MONI(9,A);
75 3      END PRINTM;

76 2      PRINT: PROCEDURE(A);
77 3          DECLARE A ADDRESS;
          /* PRINT THE STRING STARTING AT ADDRESS A UNTIL THE
          NEXT DOLLAR SIGN IS ENCOUNTERED WITH PRECEDING CRLF */
78 3          CALL CRLF;
79 3          CALL PRINTM(A);
80 3      END PRINT;

81 2      declare    mbuffer address,
82 2                  LA ADDRESS; /* CURRENT LOAD ADDRESS */
          declare head byte;

83 2      PERROR: PROCEDURE(A);
          /* PRINT ERROR MESSAGE */
84 3          DECLARE A ADDRESS;
85 3          CALL PRINT(.'ERROR: $');
86 3          CALL PRINTM(A);
87 3          CALL PRINTM(.,', LOAD ADDRESS $');
88 3          CALL PRINTADDR(LA);
89 3          CALL BOOT;
90 3      END PERROR;

91 2      diskerror: procedure;
92 3          call perror(.'DISK WRITE$');
93 3      end diskerror;

94 2      DECLARE DCNT BYTE;
```

COPYRIGHT © 1981, 1982  
DIGITAL RESEARCH  
P.O. BOX 579  
PACIFIC GROVE, CA 93950

SER. # \_\_\_\_\_

```
95 2   setdma: procedure(a);
96 3       declare a address;
97 3       call mon1 (26,a);
98 3       end setdma;

99 2   OPEN: PROCEDURE(FCB);
100 3       DECLARE FCB ADDRESS;
101 3       DCNT = MON2(15,FCB);
102 3       END OPEN;

103 2   CLOSE: PROCEDURE(FCB);
104 3       DECLARE FCB ADDRESS;
105 3       DCNT = MON2(16,FCB);
106 3       END CLOSE;

107 2   SEARCH: PROCEDURE(FCB);
108 3       DECLARE FCB ADDRESS;
109 3       DCNT = MON2(17,FCB);
110 3       END SEARCH;

111 2   SEARCHN: PROCEDURE;
112 3       DCNT = MON2(18,0);
113 3       END SEARCHN;

114 2   DELETE: PROCEDURE(FCB);
115 3       DECLARE FCB ADDRESS;
116 3       CALL MON1(19,FCB);
117 3       END DELETE;

118 2   DISKREAD: PROCEDURE(FCB) BYTE;
119 3       DECLARE FCB ADDRESS;
120 3       RETURN MON2(20,FCB);
121 3       END DISKREAD;

122 2   DISKWRITE: PROCEDURE(FCB) BYTE;
123 3       DECLARE FCB ADDRESS;
124 3       RETURN MON2(21,FCB);
125 3       END DISKWRITE;

126 2   MAKE: PROCEDURE(FCB);
127 3       DECLARE FCB ADDRESS;
128 3       DCNT = MON2(22,FCB);
129 3       END MAKE;

130 2   RENAME: PROCEDURE(FCB);
131 3       DECLARE FCB ADDRESS;
132 3       CALL MON1(23,FCB);
133 3       END RENAME;

134 2   MOVE: PROCEDURE(S,D,N);
135 3       DECLARE (S,D) ADDRESS, N BYTE,
136 3       A BASED S BYTE, B BASED D BYTE;
137 4       DO WHILE (N:=N-1) < 255;
138 4           B = A; S=S+1; D=D+1;
139 4       END;
140 4       END;
141 3       END MOVE;
```

COPYRIGHT © 1981, 1982

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # \_\_\_\_\_

```

142  2      declare char byte;

143  2      comline$error: procedure;
144  3          declare i byte;
145  3          call crlf;
146  3          do i = 1 to tbp;
147  4              call printchar (buffer(i));
148  4          end;
149  3          call printchar ('?');
150  3          call crlf;
151  3          call boot;
152  3          end comline$error;

153  2      retchar: procedure byte;
154  3          /* get another character from command tail */
155  3          if (tbp := tbp+1) <= count$command$tail then
156  3              return buffer(tbp);
157  3          else return (0dh);
158  3          end retchar;

158  2      tran: procedure(b) byte;
159  3          declare b byte;
160  3          if b < ' ' then return 0dh; /* non-graphic */
161  3          if b - 'a' < ('z' - 'a') then
162  3              b = b and 101$1111b; /* upper case */
163  3          return b;
164  3          end tran;

166  2      next$non$blank: procedure;
167  3          char=tran(retchar);
168  3          do while char= ' ';
169  4              char= tran(retchar);
170  4          end;
171  3          end next$non$blank;

172  2      MAKE$DOUBLE: PROCEDURE(H,L) ADDRESS;
173  3          /* CREATE A DOUBLE BYTE VALUE FROM TWO SINGLE BYTES */
174  3          DECLARE (H,L) BYTE;
175  3          RETURN SHL(DOUBLE(1),8) OR h;
176  3          END MAKE$DOUBLE;

176  2      delimiter: procedure byte; /* logical */

```

COPYRIGHT © 1981, 1982

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # \_\_\_\_\_

```

177 3      declare i byte;
178 3      declare del (*) byte data (0dh,'[]',' ');
179 3      do i = 0 to last(del);
180 4          if char = del(i) then return true;
182 4      end;
183 3      return false;
184 3      end delimiter;

185 2      CHECK$ONE$HEX: PROCEDURE (h) BYTE;
          /* READ ONE HEX CHARACTER FROM THE INPUT */
186 3      DECLARE H BYTE;
187 3      IF H - '0' <= 9 THEN RETURN H - '0';
189 3      IF H - 'A' > 5 THEN
190 3          return (0ffh);
191 3      RETURN H - 'A' + 10;
192 3      END CHECK$ONE$HEX;

```

```

193 2      get$num: procedure address;
194 3      declare paradd address;
195 3      paradd = 0;
196 3      char = retchar;
197 3      do while not delimiter ;
198 4          if (char:=check$one$hex(char)) = 0ffh then
199 4              call comline$error; else
200 4              paradd = paradd * 16 + char;
201 4              char = retchar;
202 4              end;

203 3      return paradd;
204 3      end get$num;

```

COPYRIGHT © 1981, 1982

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # \_\_\_\_\_

```

205 2      GETCHAR: PROCEDURE BYTE;
          /* GET NEXT CHARACTER FROM DISK BUFFER */
206 3      DECLARE I BYTE;
207 3      IF (SBP := SBP+1) <= LAST(SBUFF) THEN
208 3          RETURN SBUFF(SBP);
          /* OTHERWISE READ ANOTHER BUFFER FULL */
209 3      DO SBP = 0 TO LAST(SBUFF) BY 128;
210 4      IF (I:=DISKREAD(.SFCB)) = 0 THEN
211 4          CALL MOVE(.buffer,.SBUFF(SBP),80H); ELSE
212 4      DO;
213 5      IF I<>1 THEN CALL PERROR(,('DISK READ$'));
215 5      SBUFF(SBP) = EOF;
216 5      SBP = LAST(SBUFF);
217 5      END;
218 4      END;
219 3      SBP = 0; RETURN SBUFF(0);
221 3      END GETCHAR;
222 2      DECLARE
          STACKPOINTER LITERALLY 'STACKPTR';

```

/\* INTEL HEX FORMAT LOADER \*/

```

223 2  RELOC: PROCEDURE;
224 3      declare lnames literally '96h';
225 3      declare pedata literally '04h';
226 3      declare pidata literally '86h';
227 3      declare segdef literally '98h';
228 3      declare groupdef literally '9ah';
229 3      DECLARE ( CS, RT,K,J,jk,check$len) BYTE;
230 3      declare r1 address;
231 3      declare matched byte;
232 3      declare dlength byte;
233 3      DECLARE
          tabs address, /* temporary value */
          TA ADDRESS, /* TEMP ADDRESS */
          SA ADDRESS, /* START ADDRESS */
          FA ADDRESS, /* FINAL ADDRESS */
          NB ADDRESS, /* NUMBER OF BYTES LOADED */
          nxb byte, /* next byte in stream */
          segadjust address, /* segment adjust */
          seg$length (8) address, /* length of each segment */
          writetadd address,

          ns byte, /* number of segments in group */
          MBUFF based mbuffadr (256) BYTE,
          P BYTE;
234 3      declare high$add address;

235 3  SETMEN: PROCEDURE(8);
          /* set mbuff to b at location la */
236 4      DECLARE (B) BYTE;
237 4      if (.memory+la) > maxb then
238 4          do;
239 5          call print (('INSUFFICIENT MEMORY TO CREATE CMD FILE $'));
240 5          call boot;
241 5          end;
242 4      MBUFF(LA) = B;
243 4      END SETMEN;

244 3  zero$mem: procedure;
245 4      do while (.memory+la) < maxb and not nozero;
246 5          mbuff(la) = 0;
247 5          la = la +1;
248 5          end;
249 4      end zero$mem;

250 3  DIAGNOSE: PROCEDURE;
251 4      DECLARE N BASED TA BYTE;

252 4  NEWLINE: PROCEDURE;
253 5      CALL CRLF; CALL PRINTADDR(TA); CALL PRINTCHAR(' ');
254 5      CALL PRINTCHAR(' ');
255 5      END NEWLINE;

```

COPYRIGHT © 1981, 1982

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # \_\_\_\_\_

```

/* PRINT DIAGNOSTIC INFORMATION AT THE CONSOLE */
258 4      CALL PRINT(('LOAD ADDRESS $')); CALL PRINTADDR(TA);
260 4      CALL PRINT(('ERROR ADDRESS $')); CALL PRINTADDR(LA);

262 4      CALL PRINT(('BYTES READ:$')); CALL NEWLINE;
264 4      DO WHILE TA < LA;
265 5      IF (LOW(TA) AND OFH) = 0 THEN CALL NEWLINE;
267 5      CALL PRINTEX(MBUFF(TA)); TA=TA+1;
269 5      CALL PRINTCHAR(' ');
270 5      END;
271 4      CALL CRLF;
272 4      CALL BOOT;
273 4      END DIAGNOSE;
274 3      write$record: procedure;

275 4      call setdma(write$add);
276 4      if diskwrite(.fcba) <> 0 then call diskerror;
278 4      p = p+1;
279 4      end write$record;

280 3      empty$buffers: procedure;
281 4      write$add = .memory;
282 4      do while write$add+127 <= (.memory+fa);
283 5      call write$record;
284 5      write$add = write$add+128;
285 5      end;
286 4      call write$record;
287 4      end empty$buffers;

288 3      READCS: PROCEDURE BYTE;
289 4      /* READ BYTE WHILE COMPUTING CHECKSUM */
290 4      DECLARE B BYTE;
291 4      CS = CS + (B := getchar);
292 4      RETURN B;
293 4      END READCS;

293 3      bin$input: procedure;

294 4      do;
295 5      /* PROCESS EACH BYTE */
296 5      DO WHILE (RL := RL - 1) <> 1;
297 6      CALL SETMEM(READCS); LA = LA+1;
298 6      END;
299 5      IF LA > FA THEN FA = LA - 1;
301 5      end;

302 4      end bin$input;

```

COPYRIGHT © 1981, 1982

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # \_\_\_\_\_



```

303 3  get$laddr: procedure;
304 4      la = (make$double(readcs,readcs)*16) + readcs;
/* call printaddr (la);
   call crlf; */
305 4  end get$laddr;

306 3      compare: procedure byte;
307 4      declare k byte;
308 4      declare temp(6) byte;
309 4      do k = 0 to 5;
310 5          temp(k) = readcs;
311 5          rl=rl-1;
312 5      end;

313 4      do k = 0 to 23 by 6;
314 5          matched = true;
315 5          do j = 0 to 5;
316 6              if group(j,k) <> temp(j) then matched = false;
318 6          end;
319 5          if matched = true then go to exit;
321 5      end;
322 4      return (Offh);
323 4      exit: return ((k+1)/6);
324 4  end compare;

```

```

325 3  skipby: procedure(b);
326 4      declare (b,k) byte;
327 4      do k = 1 to b;
328 5          j = readcs;
329 5          rl=rl-1;
330 5      end;
331 4  end skipby;

```

COPYRIGHT © 1981, 1982

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # \_\_\_\_\_

```

/* INITIALIZE */
332 3  SA, FA, NB, ns = 0;
333 3  P = 0; /* PARAGRAPH COUNT */
334 3  SBUFF(0) = EOF;
335 3  fcb(nr) = 0;
336 3  segadjst = 0;
337 3  do k= 0 to 7;
338 4      seglength(k) = 0;
339 4  end;

340 3      call zerosmem;

341 3      ta=0;
342 3      la=1;
/* READ RECORDS UNTIL :00XXXX IS ENCOUNTERED */

343 3      DO FOREVER;
/* MAY BE THE END OF TAPE */

```

```

344 4      /* SET CHECK SUM TO ZERO, AND SAVE THE RECORD LENGTH */
        CS = 0;

        /* READ THE RECORD TYPE */

345 4      if (rt:=readcs) = eofile then go to fin;
347 4      rl = makedouble(readcs,readcs);
348 4      if rt = lnames then
349 4          do;
350 5          k = readcs;
351 5          rl = rl - 1;
352 5          k = 2;
353 5          do while rl < 1;
354 6              rl = rl - 1;
355 6              if (dlength := readcs) = 6 then
356 6                  do;
357 7                      if (jk := compare) < 0ffh then group$index(jk) = k;
359 7                      end; else
360 6                          call skipby(dlength);
361 6                          k = k + 1;
362 6                      end;
        /* do k = 0 to 3;
            call printhex(group$index(k));
            end;
            call crlf; */
363 5          end;
364 4      if rt = segdef then
365 4          do;
366 5          call skipby(4);
367 5          lseg$length(ns) = makedouble(readcs,readcs);
368 5          rl = rl - 2;
        /* call printaddr (lseg$length(ns));
            call crlf; */
369 5          ns = ns + 1;
370 5          end;

371 4      if rt = groupdef then
372 4          do;
373 5          nb = readcs;
374 5          rl = rl - 1;
375 5          k = 0;
376 5          do while nb < group$index(k) and (k:=k+1) <= 4;
377 6              end;
378 5          k = k-1;
379 5          call skipby(4);
380 5          do while rl < 1;
381 6              nb = readcs;
382 6              header(k).file$length = header(k).file$length + lseg$length(jk:=readcs-1);
383 6              header(k).typseq = k+1;
        /* call printhex(k);
            call printchar(20h);
            call printhex(jk);
            call printchar(20h);
            call printaddr(header(k).file$length);

```

COPYRIGHT © 1981, 1982

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # \_\_\_\_\_

```

384 6      call crlf; */
385 6      rl = rl - 2;
          end;

```

```

386 5      end;

```

```

387 4      if rt = pedata then
388 4          do;
389 5          call get$laddr;
390 5          rl = rl - 2;
391 5          call bin$input;
392 5          end;

```

```

393 4      call skipby(rl);

```

```

394 4      end;      /* end of forever loop on reading records */

```

COPYRIGHT © 1981, 1982  
 DIGITAL RESEARCH  
 P. O. BOX 579  
 PACIFIC GROVE, CA 93950

SER. # \_\_\_\_\_

```

395 3      FIN:
          do k = 0 to 7;
396 4          if header(k).file$length <> 0 then
397 4              do;
398 5                  header(k).file$length = (header(k).file$length+paragraph and 0fff0h)/16;
399 5                  if header(k).minimum$mem=0 then
400 5                      header(k).minimum$mem=header(k).file$length;
401 5                  end;
402 4              end;

```

```

          /* write the header record */
403 3          call move (.header,.buffer,128);
404 3          call setdma(.buffer);
405 3          if diskwrite(.fcha) <> 0 then call diskerror;

407 3          call empty$buffers;

```

```

          /* PRINT FINAL STATISTICS */
408 3          CALL PRINT(,('RECORDS WRITTEN ')); CALL PRINTEX(P+1);
410 3          CALL CRLF;

```

```

411 3      END RELOC;

```

```

412 2      declare seg$number byte;

```

```

413 2      ignore$filename: procedure;
414 3          tbp = 0;
415 3          char = buffer(tbp);
416 3          call next$non$blank;
417 3          do while (char:=buffer(tbp)) <> ' ';
418 4          tbp = tbp + 1;
419 4          end;

```

```

420 3      end ignore$filename;

421 2      parse$tail: procedure;
422 3          declare seg$index byte;

423 3          get$segmt: procedure byte;
          /* get the segment name */
424 4          declare ( kentry, match$flag,j, no$match) byte;
425 4          declare user$segmt(5) byte;

426 4          do j = 0 to last (user$segmt);
427 5              if delimiter then
428 5                  user$segmt(j) = ' '; else
429 5                  do;
430 6                      user$segmt(j) = char;
431 6                      char = tran(retchar);
432 6                      end;
433 5                  end;

434 4          seg$index = 0;
435 4          no$match, match$flag = true;

436 4          do while no$match and seg$index < 11;

437 5              match$flag=true;
438 5              kentry = 0;
439 5              do while match$flag and kentry <= last (segmts.name);
440 6                  if user$segmt(kentry) <> segmts(seg$index).name(kentry) then
441 6                      match$flag = false; else
442 6                      kentry = kentry +1;
443 6                      end;
444 5              if match$flag then no$match = false; else
445 5                  seg$index = seg$index +1;
446 5              end;
447 5              if no$match then seg$index = 0fff;
448 4              return seg$index;
449 4          end get$segmt;

450 4          end get$segmt;

451 4          end get$segmt;

452 3      get$switches: procedure;
453 4          do while char <> ']' and char <> cr;
454 5              call next$non$blank;
455 5              if char= 'A' then header(seg$index).absolute$add = (get$num);
456 5              else if
457 5                  char= 'M' then
458 5                  do;
459 6                      header(seg$index).minimum$num = (get$num);
460 6                      header(seg$index).typseg = seg$index+1;
461 6                      end;
462 5              else if
463 5                  char= 'X' then header(seg$index).maximum$num = (get$num);
464 5              else do;
465 6                  call comline$error;
466 6                  call boot;

```

COPYRIGHT © 1981, 1982

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # \_\_\_\_\_

```

467 6      end ;
468 5      end;

```

```

469 4      end get$switches;

```

```

470 3      do forever;
471 4          call next$non$blank;
472 4          if char = cr then return;
474 4          if get$segt = 0ffh then
475 4              do;
476 5              call comline$error;
477 5              end;
478 4          if seg$index < 8 then
479 4              do;
480 5              if char = ']' or char = cr then call comline$error;
482 5              call get$switches;
483 5              end;
484 4          else
485 4              do;
486 5              if seg$index = 8 then t8080 = true; else
487 5              do;
488 6              if seg$index = 9 then nozero = true; else
489 6              head = false;
490 6              end;
491 6              end;
492 5              end;
493 4          end;
494 3      end parse$tail;

```

COPYRIGHT © 1981, 1982

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # \_\_\_\_\_

```

/* ARRIVE HERE FROM THE SYSTEM MONITOR, READY TO READ THE HEX TAPE */

```

```

/* SET UP STACKPOINTER IN THE LOCAL AREA */

```

```

495 2  DECLARE STACK(64) ADDRESS;
496 2  SP = STACKPOINTER; STACKPOINTER = .STACK(LENGTH(STACK));
498 2  LA = 0h;
499 2  sbuffadr = .memory;
500 2  t8080 = false;
501 2  nozero = false;
502 2  head = true;

503 2  SBP = LENGTH(SBUFF);
504 2  /* SET UP THE SOURCE FILE */
505 2  CALL MOVE(.FCBA,.SFCB,33);
506 2  CALL MOVE(.(('ABS',0),.SFCB(9),4));
507 2  CALL OPEN(.SFCB);
508 2  IF DCNT = 255 THEN CALL PERROR(.(('CANNOT OPEN SOURCE$'));
509 2  CALL MOVE(.(('CMD'),.FCBA+9,3);

```

```

/* REMOVE ANY EXISTING FILE BY THIS NAME */

```

```

510 2  CALL DELETE(.FCBA);
511 2  /* THEN OPEN A NEW FILE */

```

```
511 2 CALL MAKE(.FCBA); CALL OPEN(.FCBA);
513 2 IF DCNT = 255 THEN CALL PERROR(.(('NO MORE DIRECTORY SPACE$'))); ELSE
515 2 DO;
516 3 call ignore$filename;
517 3 call parsettail;
518 3 CALL RELOC;
519 3 CALL CLOSE(.FCBA);
520 3 IF DCNT = 255 THEN CALL PERROR(.(('CANNOT CLOSE FILE$')));
522 3 END;
523 2 CALL CRLF;

524 2 CALL BOOT;
525 2 END PLM;
526 1 END;
EOF
```

COPYRIGHT © 1981, 1982

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # \_\_\_\_\_

## CROSS-REFERENCE LISTING

DEFN ADDR SIZE NAME, ATTRIBUTES, AND REFERENCES

|     |       |     |                    |                                |                         |     |     |     |     |     |     |     |     |     |  |  |  |  |  |
|-----|-------|-----|--------------------|--------------------------------|-------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|--|--|--|--|--|
| 4   | 0000H | 2   | A. . . . .         | WORD PARAMETER                 | 5                       |     |     |     |     |     |     |     |     |     |  |  |  |  |  |
| 72  | 0004H | 2   | A. . . . .         | WORD PARAMETER AUTOMATIC       |                         |     |     |     |     |     |     |     |     |     |  |  |  |  |  |
| 95  | 0004H | 2   | A. . . . .         | WORD PARAMETER AUTOMATIC       |                         |     |     |     |     |     |     |     |     |     |  |  |  |  |  |
| 67  | 0004H | 2   | A. . . . .         | WORD PARAMETER AUTOMATIC       |                         |     |     |     |     |     |     |     |     |     |  |  |  |  |  |
| 76  | 0004H | 2   | A. . . . .         | WORD PARAMETER AUTOMATIC       |                         |     |     |     |     |     |     |     |     |     |  |  |  |  |  |
| 135 | 0000H | 1   | A. . . . .         | BYTE BASED(S)                  | 137                     |     |     |     |     |     |     |     |     |     |  |  |  |  |  |
| 83  | 0004H | 2   | A. . . . .         | WORD PARAMETER AUTOMATIC       |                         |     |     |     |     |     |     |     |     |     |  |  |  |  |  |
| 7   | 0000H | 2   | A. . . . .         | WORD PARAMETER                 | 8                       |     |     |     |     |     |     |     |     |     |  |  |  |  |  |
| 18  | 0003H | 2   | ABSOLUTEADD. . . . | WORD MEMBER(HEADER)            |                         | 456 |     |     |     |     |     |     |     |     |  |  |  |  |  |
| 135 | 0000H | 1   | B. . . . .         | BYTE BASED(D)                  | 137                     |     |     |     |     |     |     |     |     |     |  |  |  |  |  |
| 158 | 0004H | 1   | B. . . . .         | BYTE PARAMETER AUTOMATIC       |                         |     | 159 | 160 | 162 | 163 | 164 |     |     |     |  |  |  |  |  |
| 325 | 0004H | 1   | B. . . . .         | BYTE PARAMETER AUTOMATIC       |                         |     | 326 | 327 |     |     |     |     |     |     |  |  |  |  |  |
| 235 | 0004H | 1   | B. . . . .         | BYTE PARAMETER AUTOMATIC       |                         |     | 236 | 242 |     |     |     |     |     |     |  |  |  |  |  |
| 289 | 05DEH | 1   | B. . . . .         | BYTE                           | 290 291                 |     |     |     |     |     |     |     |     |     |  |  |  |  |  |
| 62  | 0004H | 1   | B. . . . .         | BYTE PARAMETER AUTOMATIC       |                         |     | 63  | 64  | 65  |     |     |     |     |     |  |  |  |  |  |
| 293 | 0790H | 43  | BININPUT . . . . . | PROCEDURE STACK=002EH          |                         |     | 391 |     |     |     |     |     |     |     |  |  |  |  |  |
| 11  | 0000H | 14  | BOOT . . . . .     | PROCEDURE STACK=0008H          |                         |     | 89  | 151 | 240 | 272 | 466 | 524 |     |     |  |  |  |  |  |
| 22  |       |     | B. SIZE. . . . .   | LITERALLY                      | 22                      |     |     |     |     |     |     |     |     |     |  |  |  |  |  |
| 2   | 0000H | 128 | BUFFA. . . . .     | BYTE ARRAY(128) EXTERNAL(2)    |                         |     |     | 21  | 24  |     |     |     |     |     |  |  |  |  |  |
| 21  | 0000H | 128 | BUFFER . . . . .   | BYTE ARRAY(128) EXTERNAL(2) AT |                         |     |     | 147 | 155 | 211 | 403 | 404 | 415 | 417 |  |  |  |  |  |
| 142 | 05CFH | 1   | CHAR . . . . .     | BYTE                           | 167 168 169 180         | 196 | 198 | 200 | 201 | 415 | 417 | 430 | 431 |     |  |  |  |  |  |
|     |       |     |                    |                                | 453 455 457 462 472 480 |     |     |     |     |     |     |     |     |     |  |  |  |  |  |
| 43  | 0004H | 1   | CHAR . . . . .     | BYTE PARAMETER AUTOMATIC       |                         |     | 49  | 50  |     |     |     |     |     |     |  |  |  |  |  |
| 229 | 05D8H | 1   | CHECKLEN . . . . . | BYTE                           |                         |     |     |     |     |     |     |     |     |     |  |  |  |  |  |
| 185 | 0366H | 36  | CHECKONEHEX. . . . | PROCEDURE BYTE STACK=0004H     |                         |     |     | 198 |     |     |     |     |     |     |  |  |  |  |  |
| 103 | 01E9H | 19  | CLOSE. . . . .     | PROCEDURE STACK=000AH          |                         |     | 519 |     |     |     |     |     |     |     |  |  |  |  |  |
| 143 | 0297H | 51  | COMLINEERROR . . . | PROCEDURE STACK=0012H          |                         |     | 199 | 465 | 476 | 481 |     |     |     |     |  |  |  |  |  |
| 306 | 07DCH | 138 | COMPARE. . . . .   | PROCEDURE BYTE STACK=002EH     |                         |     | 357 |     |     |     |     |     |     |     |  |  |  |  |  |
| 3   | 0000H | 38  | COPYRIGHT. . . . . | BYTE ARRAY(38) DATA            |                         |     |     |     |     |     |     |     |     |     |  |  |  |  |  |
| 24  | 0000H | 1   | COUNTCOMHANDTAIL . | BYTE EXTERNAL(2) AT            | 154                     |     |     |     |     |     |     |     |     |     |  |  |  |  |  |
| 26  |       |     | CR . . . . .       | LITERALLY                      | 53 453 472              | 480 |     |     |     |     |     |     |     |     |  |  |  |  |  |
| 52  | 0113H | 17  | CRLF . . . . .     | PROCEDURE STACK=000EH          |                         |     | 78  | 145 | 150 | 253 | 271 | 410 | 523 |     |  |  |  |  |  |
| 229 | 05D3H | 1   | CS . . . . .       | BYTE                           | 290 344                 |     |     |     |     |     |     |     |     |     |  |  |  |  |  |
| 134 | 0006H | 2   | D. . . . .         | WORD PARAMETER AUTOMATIC       |                         |     | 135 | 137 | 139 |     |     |     |     |     |  |  |  |  |  |
| 21  |       |     | DBUFF. . . . .     | LITERALLY                      |                         |     |     |     |     |     |     |     |     |     |  |  |  |  |  |
| 94  | 05CEH | 1   | DCNT . . . . .     | BYTE                           | 101 105 109 112         | 128 | 507 | 513 | 520 |     |     |     |     |     |  |  |  |  |  |
| 178 | 0026H | 5   | DEL. . . . .       | BYTE ARRAY(5) DATA             | 179                     | 180 |     |     |     |     |     |     |     |     |  |  |  |  |  |
| 114 | 0221H | 16  | DELETE . . . . .   | PROCEDURE STACK=000AH          |                         |     | 510 |     |     |     |     |     |     |     |  |  |  |  |  |
| 176 | 033AH | 44  | DELIMITER. . . . . | PROCEDURE BYTE STACK=0002H     |                         |     | 197 | 427 |     |     |     |     |     |     |  |  |  |  |  |
| 20  |       |     | DFCBA. . . . .     | LITERALLY                      |                         |     |     |     |     |     |     |     |     |     |  |  |  |  |  |
| 250 | 05COH | 90  | DIAGNOSE . . . . . | PROCEDURE STACK=0024H          |                         |     |     |     |     |     |     |     |     |     |  |  |  |  |  |
| 91  | 01BAH | 12  | DISKERROR. . . . . | PROCEDURE STACK=0026H          |                         |     | 277 | 406 |     |     |     |     |     |     |  |  |  |  |  |
| 118 | 0231H | 16  | DISKREAD . . . . . | PROCEDURE BYTE STACK=000AH     |                         |     |     | 210 |     |     |     |     |     |     |  |  |  |  |  |
| 122 | 0241H | 16  | DISKWRITE. . . . . | PROCEDURE BYTE STACK=000AH     |                         |     | 276 | 405 |     |     |     |     |     |     |  |  |  |  |  |
| 232 | 05DAH | 1   | DLENGTH. . . . .   | BYTE                           | 355 360                 |     |     |     |     |     |     |     |     |     |  |  |  |  |  |
|     |       |     | DOUBLE . . . . .   | BUILTIN                        | 174                     |     |     |     |     |     |     |     |     |     |  |  |  |  |  |
| 230 | 0753H | 43  | EMPTYBUFFERS . . . | PROCEDURE STACK=002EH          |                         |     | 407 |     |     |     |     |     |     |     |  |  |  |  |  |
| 22  |       |     | EOFIL . . . . .    | LITERALLY                      | 215 334 345             |     |     |     |     |     |     |     |     |     |  |  |  |  |  |
| 2   |       |     | EX . . . . .       | LITERALLY                      |                         |     |     |     |     |     |     |     |     |     |  |  |  |  |  |

COPYRIGHT © 1981, 1982  
DIGITAL RESEARCH  
P.O. BOX 579  
PACIFIC GROVE, CA 93950  
SER. #

|     |       |                        |                               |     |     |     |     |     |     |     |     |     |     |     |     |  |  |  |  |
|-----|-------|------------------------|-------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|--|--|--|--|
| 323 | 0856H | EXIT . . . . .         | LABEL                         | 320 |     |     |     |     |     |     |     |     |     |     |     |  |  |  |  |
| 7   | 0000H | 1 F. . . . .           | BYTE PARAMETER                |     | 8   |     |     |     |     |     |     |     |     |     |     |  |  |  |  |
| 4   | 0000H | 1 F. . . . .           | BYTE PARAMETER                |     | 5   |     |     |     |     |     |     |     |     |     |     |  |  |  |  |
| 233 | 0034H | 2 FA . . . . .         | WORD                          | 282 | 299 | 300 | 332 |     |     |     |     |     |     |     |     |  |  |  |  |
| 26  |       | FALSE. . . . .         | LITERALLY                     | 183 | 317 | 441 | 445 | 490 | 500 | 501 |     |     |     |     |     |  |  |  |  |
| 126 | 0004H | 2 FCB. . . . .         | WORD PARAMETER AUTOMATIC      |     |     |     | 127 | 128 |     |     |     |     |     |     |     |  |  |  |  |
| 20  | 0000H | 33 FCB. . . . .        | BYTE ARRAY(33) EXTERNAL(1) AT |     |     |     |     | 335 |     |     |     |     |     |     |     |  |  |  |  |
| 130 | 0004H | 2 FCB. . . . .         | WORD PARAMETER AUTOMATIC      |     |     |     | 131 | 132 |     |     |     |     |     |     |     |  |  |  |  |
| 122 | 0004H | 2 FCB. . . . .         | WORD PARAMETER AUTOMATIC      |     |     |     | 123 | 124 |     |     |     |     |     |     |     |  |  |  |  |
| 114 | 0004H | 2 FCB. . . . .         | WORD PARAMETER AUTOMATIC      |     |     |     | 115 | 116 |     |     |     |     |     |     |     |  |  |  |  |
| 107 | 0004H | 2 FCB. . . . .         | WORD PARAMETER AUTOMATIC      |     |     |     | 108 | 109 |     |     |     |     |     |     |     |  |  |  |  |
| 103 | 0004H | 2 FCB. . . . .         | WORD PARAMETER AUTOMATIC      |     |     |     | 104 | 105 |     |     |     |     |     |     |     |  |  |  |  |
| 79  | 0004H | 2 FCB. . . . .         | WORD PARAMETER AUTOMATIC      |     |     |     | 100 | 101 |     |     |     |     |     |     |     |  |  |  |  |
| 118 | 0004H | 2 FCB. . . . .         | WORD PARAMETER AUTOMATIC      |     |     |     | 119 | 120 |     |     |     |     |     |     |     |  |  |  |  |
| 2   | 0004H | 33 FCDA . . . . .      | BYTE ARRAY(33) EXTERNAL(1)    |     |     |     | 20  | 276 | 405 | 504 | 509 | 510 | 511 |     |     |  |  |  |  |
|     |       |                        |                               | 512 | 519 |     |     |     |     |     |     |     |     |     |     |  |  |  |  |
| 18  | 0001H | 2 FILELENGTH . . . .   | WORD MEMBER(HEADER)           |     | 382 | 396 | 398 | 400 |     |     |     |     |     |     |     |  |  |  |  |
| 395 | 05EBH | FIN. . . . .           | LABEL                         | 346 |     |     |     |     |     |     |     |     |     |     |     |  |  |  |  |
| 26  |       | FOREVER. . . . .       | LITERALLY                     | 343 | 470 |     |     |     |     |     |     |     |     |     |     |  |  |  |  |
| 205 | 03CFH | 124 GETCHAR. . . . .   | PROCEDURE BYTE STACK=0026H    |     |     |     |     | 290 |     |     |     |     |     |     |     |  |  |  |  |
| 303 | 078BH | 33 GETLADDR . . . . .  | PROCEDURE STACK=0030H         |     | 389 |     |     |     |     |     |     |     |     |     |     |  |  |  |  |
| 193 | 038AH | 69 GETNUM . . . . .    | PROCEDURE WORD STACK=0016H    |     |     |     | 456 | 459 | 463 |     |     |     |     |     |     |  |  |  |  |
| 423 | 0912H | 201 GETSEGNT . . . . . | PROCEDURE BYTE STACK=0008H    |     |     |     | 474 |     |     |     |     |     |     |     |     |  |  |  |  |
| 452 | 09DBH | 145 GETSWITCHES. . . . | PROCEDURE STACK=001AH         |     | 482 |     |     |     |     |     |     |     |     |     |     |  |  |  |  |
| 15  | 0105H | 24 GROUP. . . . .      | BYTE ARRAY(24) INITIAL        |     | 316 |     |     |     |     |     |     |     |     |     |     |  |  |  |  |
| 228 |       | GROUPDEF . . . . .     | LITERALLY                     | 371 |     |     |     |     |     |     |     |     |     |     |     |  |  |  |  |
| 16  | 011DH | 4 GROUPINDEX . . . .   | BYTE ARRAY(4) INITIAL         |     | 358 | 376 |     |     |     |     |     |     |     |     |     |  |  |  |  |
| 172 | 0006H | 1 H. . . . .           | BYTE PARAMETER AUTOMATIC      |     | 173 | 174 |     |     |     |     |     |     |     |     |     |  |  |  |  |
| 185 | 0004H | 1 H. . . . .           | BYTE PARAMETER AUTOMATIC      |     | 186 | 187 | 188 | 189 | 191 |     |     |     |     |     |     |  |  |  |  |
| 82  | 05CDH | 1 HEAD . . . . .       | BYTE                          | 490 | 502 |     |     |     |     |     |     |     |     |     |     |  |  |  |  |
| 18  | 0121H | 135 HEADER . . . . .   | STRUCTURE ARRAY(15) INITIAL   |     |     |     | 382 | 383 | 396 | 398 | 399 | 400 | 403 |     |     |  |  |  |  |
|     |       |                        |                               | 456 | 459 | 460 | 463 |     |     |     |     |     |     |     |     |  |  |  |  |
|     |       | HIGH . . . . .         | BUILTIN                       | 39  |     |     |     |     |     |     |     |     |     |     |     |  |  |  |  |
| 234 | 004CH | 2 HIGHADD. . . . .     | WORD                          |     |     |     |     |     |     |     |     |     |     |     |     |  |  |  |  |
| 28  | 05CCH | 1 I. . . . .           | BYTE                          | 29  | 30  | 31  | 32  | 33  | 34  | 35  | 36  | 37  | 38  | 39  | 40  |  |  |  |  |
|     |       |                        |                               | 41  | 42  | 43  | 44  | 45  | 46  |     |     |     |     |     |     |  |  |  |  |
| 177 | 05D1H | 1 I. . . . .           | BYTE                          | 179 | 180 |     |     |     |     |     |     |     |     |     |     |  |  |  |  |
| 144 | 05D0H | 1 I. . . . .           | BYTE                          | 146 | 147 |     |     |     |     |     |     |     |     |     |     |  |  |  |  |
| 206 | 05D2H | 1 I. . . . .           | BYTE                          | 210 | 213 |     |     |     |     |     |     |     |     |     |     |  |  |  |  |
| 413 | 088AH | 45 IGNOREFILENAME . .  | PROCEDURE STACK=000CH         |     |     |     |     | 516 |     |     |     |     |     |     |     |  |  |  |  |
| 424 | 05EBH | 1 J. . . . .           | BYTE                          | 426 | 428 | 430 |     |     |     |     |     |     |     |     |     |  |  |  |  |
| 229 | 05D6H | 1 J. . . . .           | BYTE                          | 315 | 316 | 328 |     |     |     |     |     |     |     |     |     |  |  |  |  |
| 229 | 05D7H | 1 JK . . . . .         | BYTE                          | 357 | 358 | 382 |     |     |     |     |     |     |     |     |     |  |  |  |  |
| 326 | 05E6H | 1 K. . . . .           | BYTE                          | 327 |     |     |     |     |     |     |     |     |     |     |     |  |  |  |  |
| 307 | 05DFH | 1 K. . . . .           | BYTE                          | 309 | 310 | 313 | 316 | 323 |     |     |     |     |     |     |     |  |  |  |  |
| 229 | 05D5H | 1 K. . . . .           | BYTE                          | 337 | 338 | 350 | 352 | 358 | 361 | 375 | 376 | 378 | 382 | 383 | 395 |  |  |  |  |
|     |       |                        |                               | 396 | 398 | 399 | 400 |     |     |     |     |     |     |     |     |  |  |  |  |
| 424 | 05E9H | 1 KENTRY . . . . .     | BYTE                          | 438 | 439 | 440 | 442 |     |     |     |     |     |     |     |     |  |  |  |  |
| 172 | 0004H | 1 L. . . . .           | BYTE PARAMETER AUTOMATIC      |     | 173 | 174 |     |     |     |     |     |     |     |     |     |  |  |  |  |
| 81  | 0028H | 2 LA . . . . .         | WORD                          | 88  | 237 | 242 | 245 | 246 | 247 | 261 | 264 | 297 | 299 | 300 | 304 |  |  |  |  |
|     |       |                        |                               | 342 | 498 |     |     |     |     |     |     |     |     |     |     |  |  |  |  |
|     |       | LAST . . . . .         | BUILTIN                       | 179 | 207 | 209 | 216 | 426 | 439 |     |     |     |     |     |     |  |  |  |  |
|     |       | LENGTH . . . . .       | BUILTIN                       | 497 | 503 |     |     |     |     |     |     |     |     |     |     |  |  |  |  |
| 26  |       | LF . . . . .           | LITERALLY                     | 54  |     |     |     |     |     |     |     |     |     |     |     |  |  |  |  |
| 1   | 0000H | LHCMD. . . . .         | PROCEDURE STACK=0000H         |     |     |     |     |     |     |     |     |     |     |     |     |  |  |  |  |
| 224 |       | LNAME. . . . .         | LITERALLY                     | 348 |     |     |     |     |     |     |     |     |     |     |     |  |  |  |  |
|     |       | LOW. . . . .           | BUILTIN                       | 70  | 265 |     |     |     |     |     |     |     |     |     |     |  |  |  |  |

COPYRIGHT © 1981, 1982

DIGITAL RESEARCH

P.O. BOX 579

PACIFIC GROVE, CA 93950

SER =



|     |       |      |                      |                                        |     |     |     |     |     |     |     |     |     |     |     |     |  |  |  |  |
|-----|-------|------|----------------------|----------------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|--|--|--|--|
| 17  | 0002H | 32   | LSEGLNGTH . . . .    | WORD ARRAY(16) INITIAL                 | 367 | 382 |     |     |     |     |     |     |     |     |     |     |  |  |  |  |
| 251 | 0000H | 1    | M. . . . .           | BYTE BASED(TA)                         |     |     |     |     |     |     |     |     |     |     |     |     |  |  |  |  |
| 126 | 0251H | 19   | MAKE . . . . .       | PROCEDURE STACK=000AH                  | 511 |     |     |     |     |     |     |     |     |     |     |     |  |  |  |  |
| 172 | 0323H | 23   | MAKEDOUBLE . . . .   | PROCEDURE WORD STACK=0006H             |     | 304 | 347 | 367 |     |     |     |     |     |     |     |     |  |  |  |  |
| 231 | 05D9H | 1    | MATCHED. . . . .     | BYTE                                   | 314 | 317 | 319 |     |     |     |     |     |     |     |     |     |  |  |  |  |
| 424 | 05EAH | 1    | MATCHFLAG. . . . .   | BYTE                                   | 435 | 437 | 439 | 441 | 444 |     |     |     |     |     |     |     |  |  |  |  |
| 2   | 0000H | 2    | MAXB . . . . .       | WORD EXTERNAL(0)                       | 237 | 245 |     |     |     |     |     |     |     |     |     |     |  |  |  |  |
| 18  | 0007H | 2    | MAXIMUMMEM . . . .   | WORD MEMBER(HEADER)                    | 463 |     |     |     |     |     |     |     |     |     |     |     |  |  |  |  |
| 233 | 0000H | 256  | MBUFF. . . . .       | BYTE BASED(MBUFFADR) ARRAY(256)        |     | 242 | 246 | 267 |     |     |     |     |     |     |     |     |  |  |  |  |
| 81  | 0026H | 2    | MBUFFADR . . . . .   | WORD                                   | 233 | 242 | 246 | 267 | 499 |     |     |     |     |     |     |     |  |  |  |  |
|     | 0000H |      | MEMORY . . . . .     | BYTE ARRAY(0)                          | 237 | 245 | 281 | 282 | 499 |     |     |     |     |     |     |     |  |  |  |  |
| 18  | 0005H | 2    | MINIMUMMEM . . . .   | WORD MEMBER(HEADER)                    | 399 | 400 | 459 |     |     |     |     |     |     |     |     |     |  |  |  |  |
| 4   | 0000H |      | MON1 . . . . .       | PROCEDURE EXTERNAL(3) STACK=0000H      |     | 12  | 50  | 74  | 97  | 116 | 132 |     |     |     |     |     |  |  |  |  |
| 7   | 0000H |      | MON2 . . . . .       | PROCEDURE BYTE EXTERNAL(4) STACK=0000H |     | 101 | 105 | 109 | 112 | 120 |     |     |     |     |     |     |  |  |  |  |
|     |       |      |                      | 124 128                                |     |     |     |     |     |     |     |     |     |     |     |     |  |  |  |  |
| 134 | 0274H | 37   | MOVE . . . . .       | PROCEDURE STACK=0008H                  | 211 | 403 | 504 | 505 | 509 |     |     |     |     |     |     |     |  |  |  |  |
| 53  | 0004H | 1    | N. . . . .           | BYTE PARAMETER AUTOMATIC               | 57  | 58  | 59  | 60  |     |     |     |     |     |     |     |     |  |  |  |  |
| 134 | 0004H | 1    | N. . . . .           | BYTE PARAMETER AUTOMATIC               | 135 | 136 |     |     |     |     |     |     |     |     |     |     |  |  |  |  |
| 14  | 0000H | 5    | NAME . . . . .       | BYTE ARRAY(5) MEMBER(SEGTS)            | 439 | 440 |     |     |     |     |     |     |     |     |     |     |  |  |  |  |
| 233 | 0036H | 2    | NB . . . . .         | WORD                                   | 332 | 373 | 376 | 381 |     |     |     |     |     |     |     |     |  |  |  |  |
| 252 | 071AH | 27   | NEWLINE. . . . .     | PROCEDURE STACK=0020H                  | 263 | 266 |     |     |     |     |     |     |     |     |     |     |  |  |  |  |
| 166 | 030DH | 22   | NEXTNONBLANK . . . . | PROCEDURE STACK=0008H                  | 416 | 454 | 471 |     |     |     |     |     |     |     |     |     |  |  |  |  |
| 424 | 05ECH | 1    | NOMATCH. . . . .     | BYTE                                   | 435 | 436 | 445 | 448 |     |     |     |     |     |     |     |     |  |  |  |  |
| 25  | 05CBH | 1    | NOZERO . . . . .     | BYTE                                   | 245 | 489 | 501 |     |     |     |     |     |     |     |     |     |  |  |  |  |
| 2   |       |      | NR . . . . .         | LITERALLY                              | 335 |     |     |     |     |     |     |     |     |     |     |     |  |  |  |  |
| 233 | 05DCH | 1    | NS . . . . .         | BYTE                                   | 332 | 367 | 369 |     |     |     |     |     |     |     |     |     |  |  |  |  |
| 233 | 05D8H | 1    | NXB. . . . .         | BYTE                                   |     |     |     |     |     |     |     |     |     |     |     |     |  |  |  |  |
| 99  | 01D6H | 19   | OPEN . . . . .       | PROCEDURE STACK=000AH                  |     |     |     |     |     |     |     |     |     |     |     |     |  |  |  |  |
| 233 | 05DDH | 1    | P. . . . .           | BYTE                                   | 278 | 333 | 409 |     |     |     |     |     |     |     |     |     |  |  |  |  |
| 194 | 002AH | 2    | PARADD . . . . .     | WORD                                   | 195 | 200 | 203 |     |     |     |     |     |     |     |     |     |  |  |  |  |
| 2   |       |      | PARAGRAPH. . . . .   | LITERALLY                              | 398 |     |     |     |     |     |     |     |     |     |     |     |  |  |  |  |
| 421 | 08B7H | 91   | PARSETAIL. . . . .   | PROCEDURE STACK=001EH                  |     |     |     |     |     |     |     |     |     |     |     |     |  |  |  |  |
| 27  | 00C2H | 62   | PATCH. . . . .       | PROCEDURE STACK=0002H                  |     |     |     |     |     |     |     |     |     |     |     |     |  |  |  |  |
| 225 |       |      | PEDATA . . . . .     | LITERALLY                              | 387 |     |     |     |     |     |     |     |     |     |     |     |  |  |  |  |
| 93  | 0195H | 37   | PEEROR . . . . .     | PROCEDURE STACK=0022H                  |     |     |     |     |     |     |     |     |     |     |     |     |  |  |  |  |
| 226 |       |      | PIDATA . . . . .     | LITERALLY                              |     |     |     |     |     |     |     |     |     |     |     |     |  |  |  |  |
| 19  | 000EH | 180  | PLM. . . . .         | PROCEDURE PUBLIC STACK=0038H           |     |     |     |     |     |     |     |     |     |     |     |     |  |  |  |  |
| 76  | 0185H | 16   | PRINT. . . . .       | PROCEDURE STACK=0014H                  | 85  | 239 | 258 | 260 | 262 | 408 |     |     |     |     |     |     |  |  |  |  |
| 67  | 015EH | 23   | PRINTADDR. . . . .   | PROCEDURE STACK=001CH                  | 98  | 254 | 259 | 261 |     |     |     |     |     |     |     |     |  |  |  |  |
| 48  | 0100H | 19   | PRINTCHAR. . . . .   | PROCEDURE STACK=000AH                  | 53  | 54  | 59  | 60  | 147 | 149 | 255 | 256 |     |     |     |     |  |  |  |  |
|     |       |      |                      | 269                                    |     |     |     |     |     |     |     |     |     |     |     |     |  |  |  |  |
| 62  | 0143H | 27   | PRINTHEX . . . . .   | PROCEDURE STACK=0016H                  | 69  | 70  | 267 | 409 |     |     |     |     |     |     |     |     |  |  |  |  |
| 72  | 0175H | 16   | PRINTN . . . . .     | PROCEDURE STACK=000AH                  | 79  | 86  | 87  |     |     |     |     |     |     |     |     |     |  |  |  |  |
| 56  | 0124H | 31   | PRINTNIB . . . . .   | PROCEDURE STACK=0010H                  | 64  | 65  |     |     |     |     |     |     |     |     |     |     |  |  |  |  |
| 288 | 077EH | 18   | READCS . . . . .     | PROCEDURE BYTE STACK=002AH             | 296 | 304 | 310 | 328 | 345 | 347 | 350 |     |     |     |     |     |  |  |  |  |
|     |       |      |                      | 355 367 373 381 382                    |     |     |     |     |     |     |     |     |     |     |     |     |  |  |  |  |
| 223 | 044BH | 538  | RELOC. . . . .       | PROCEDURE STACK=0034H                  | 518 |     |     |     |     |     |     |     |     |     |     |     |  |  |  |  |
| 130 | 0264H | 16   | RENAME . . . . .     | PROCEDURE STACK=000AH                  |     |     |     |     |     |     |     |     |     |     |     |     |  |  |  |  |
| 153 | 02CCH | 32   | RETCAR. . . . .      | PROCEDURE BYTE STACK=0002H             | 167 | 169 | 196 | 201 | 431 |     |     |     |     |     |     |     |  |  |  |  |
| 22  | 05C9H | 1    | RFLAG. . . . .       | BYTE                                   |     |     |     |     |     |     |     |     |     |     |     |     |  |  |  |  |
| 230 | 002CH | 2    | RL . . . . .         | WORD                                   | 295 | 311 | 329 | 347 | 351 | 353 | 354 | 368 | 374 | 380 | 384 | 390 |  |  |  |  |
|     |       |      |                      | 393                                    |     |     |     |     |     |     |     |     |     |     |     |     |  |  |  |  |
| 229 | 05D4H | 1    | RT . . . . .         | BYTE                                   | 345 | 348 | 364 | 371 | 387 |     |     |     |     |     |     |     |  |  |  |  |
| 134 | 0008H | 2    | S. . . . .           | WORD PARAMETER AUTOMATIC               | 135 | 137 | 138 |     |     |     |     |     |     |     |     |     |  |  |  |  |
| 233 | 0032H | 2    | SA . . . . .         | WORD                                   | 332 |     |     |     |     |     |     |     |     |     |     |     |  |  |  |  |
| 22  | 0022H | 2    | SBP. . . . .         | WORD                                   | 207 | 208 | 209 | 211 | 215 | 216 | 219 | 503 |     |     |     |     |  |  |  |  |
| 22  | 01C9H | 1024 | SHUFF. . . . .       | BYTE ARRAY(1024)                       | 207 | 208 | 209 | 211 | 215 | 216 | 219 | 503 |     |     |     |     |  |  |  |  |

506 512 COPYRIGHT © 1981, 1982  
DIGITAL RESEARCH  
P.O. BOX 578  
517 PACIFIC GROVE, CA 93950  
SER. =  
92 214 508 514 521

|     |       |     |                        |                             |     |     |     |     |     |     |     |     |     |     |     |     |  |  |  |
|-----|-------|-----|------------------------|-----------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|--|--|--|
| 107 | 01FCH | 19  | SEARCH . . . . .       | PROCEDURE STACK=000AH       |     |     |     |     |     |     |     |     |     |     |     |     |  |  |  |
| 111 | 020FH | 18  | SEARCHN. . . . .       | PROCEDURE STACK=0008H       |     |     |     |     |     |     |     |     |     |     |     |     |  |  |  |
| 233 | 003BH | 2   | SEGADJST . . . . .     | WORD                        | 336 |     |     |     |     |     |     |     |     |     |     |     |  |  |  |
| 227 |       |     | SEGDEF . . . . .       | LITERALLY                   | 354 |     |     |     |     |     |     |     |     |     |     |     |  |  |  |
| 422 | 05EBH | 1   | SEGINDEX . . . . .     | BYTE                        | 434 | 436 | 440 | 446 | 449 | 450 | 456 | 459 | 460 | 463 | 478 | 485 |  |  |  |
|     |       |     |                        |                             | 488 |     |     |     |     |     |     |     |     |     |     |     |  |  |  |
| 233 | 003AH | 16  | SEGLENGTH. . . . .     | WORD ARRAY(8)               | 338 |     |     |     |     |     |     |     |     |     |     |     |  |  |  |
| 14  | 00CEH | 55  | SEGNTS . . . . .       | STRUCTURE ARRAY(11) INITIAL |     |     |     |     |     | 439 | 440 |     |     |     |     |     |  |  |  |
| 412 | 05E7H | 1   | SEGNUMBER. . . . .     | BYTE                        |     |     |     |     |     |     |     |     |     |     |     |     |  |  |  |
| 95  | 01C6H | 16  | SETDMA . . . . .       | PROCEDURE STACK=000AH       |     |     |     |     |     | 275 | 404 |     |     |     |     |     |  |  |  |
| 235 | 0665H | 42  | SETMEN . . . . .       | PROCEDURE STACK=001AH       |     |     |     |     |     | 296 |     |     |     |     |     |     |  |  |  |
| 22  | 01ABH | 33  | SFCB . . . . .         | BYTE ARRAY(33)              |     | 210 | 504 | 505 | 506 |     |     |     |     |     |     |     |  |  |  |
|     |       |     | SHL. . . . .           | BUILTIN                     | 174 |     |     |     |     |     |     |     |     |     |     |     |  |  |  |
|     |       |     | SHR. . . . .           | BUILTIN                     | 64  |     |     |     |     |     |     |     |     |     |     |     |  |  |  |
| 325 | 0866H | 36  | SKIPBY . . . . .       | PROCEDURE STACK=0030H       |     |     |     |     |     | 360 | 366 | 370 | 393 |     |     |     |  |  |  |
| 10  | 0000H | 2   | SP . . . . .           | WORD                        | 496 |     |     |     |     |     |     |     |     |     |     |     |  |  |  |
| 495 | 004EH | 128 | STACK. . . . .         | WORD ARRAY(64)              |     | 497 |     |     |     |     |     |     |     |     |     |     |  |  |  |
| 222 |       |     | STACKPOINTER . . . . . | LITERALLY                   | 496 |     |     |     |     |     |     |     |     |     |     |     |  |  |  |
|     |       |     | STACKPTR . . . . .     | BUILTIN                     | 496 | 497 |     |     |     |     |     |     |     |     |     |     |  |  |  |
| 25  | 05CAH | 1   | T8080. . . . .         | BYTE                        | 486 | 500 |     |     |     |     |     |     |     |     |     |     |  |  |  |
| 233 | 0030H | 2   | TA . . . . .           | WORD                        | 251 | 254 | 259 | 264 | 265 | 267 | 268 | 341 |     |     |     |     |  |  |  |
| 233 | 002EH | 2   | TABS . . . . .         | WORD                        |     |     |     |     |     |     |     |     |     |     |     |     |  |  |  |
| 23  | 0024H | 2   | TBP. . . . .           | WORD                        | 146 | 154 | 155 | 414 | 415 | 417 | 418 |     |     |     |     |     |  |  |  |
| 308 | 05E0H | 6   | TEMP . . . . .         | BYTE ARRAY(6)               |     | 310 | 316 |     |     |     |     |     |     |     |     |     |  |  |  |
| 150 | 02ECH | 33  | TRAN . . . . .         | PROCEDURE BYTE STACK=0004H  |     |     |     |     |     | 167 | 169 | 431 |     |     |     |     |  |  |  |
| 26  |       |     | TRUE . . . . .         | LITERALLY                   | 181 | 314 | 319 | 343 | 435 | 437 | 470 | 486 | 489 | 502 |     |     |  |  |  |
| 18  | 0000H | 1   | TYPSEG . . . . .       | BYTE MEMBER(HEADER)         |     | 383 | 460 |     |     |     |     |     |     |     |     |     |  |  |  |
| 425 | 05EDH | 5   | USERSEGNT. . . . .     | BYTE ARRAY(5)               |     | 426 | 428 | 430 | 440 |     |     |     |     |     |     |     |  |  |  |
| 26  |       |     | WHAT . . . . .         | LITERALLY                   |     |     |     |     |     |     |     |     |     |     |     |     |  |  |  |
| 233 | 004AH | 2   | WRITEADD . . . . .     | WORD                        | 275 | 281 | 282 | 284 |     |     |     |     |     |     |     |     |  |  |  |
| 274 | 0735H | 30  | WRITERECORD. . . . .   | PROCEDURE STACK=002AH       |     |     |     |     |     | 283 | 286 |     |     |     |     |     |  |  |  |
| 244 | 06BFH | 49  | ZEROMEN. . . . .       | PROCEDURE STACK=0002H       |     |     |     |     |     | 340 |     |     |     |     |     |     |  |  |  |

## MODULE INFORMATION:

CODE AREA SIZE = 0A6CH 2668D  
 CONSTANT AREA SIZE = 00FFH 255D  
 VARIABLE AREA SIZE = 05F2H 1522D  
 MAXIMUM STACK SIZE = 003BH 56D  
 749 LINES READ  
 0 PROGRAM ERROR(S)

END OF PL/M-86 COMPILATION

COPYRIGHT © 1981, 1982

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # \_\_\_\_\_